

## SYSTEM AND METHOD FOR CHANGING HDL SPECIFICATION AND STORAGE MEDIUM RECORDING ITS CONTROL PROGRAM

Patent Number: JP2000148804  
Publication date: 2000-05-30  
Inventor(s): IIZUKA TAKUYA  
Applicant(s):: NEC ENG LTD  
Requested Patent: ☐ JP2000148804 (JP00148804)  
Application Number: JP19980315723 19981106  
Priority Number(s):  
IPC Classification: G06F17/50  
EC Classification:  
Equivalents:

### Abstract

**PROBLEM TO BE SOLVED:** To provide an HDL(hardware description language) specification changing system which can easily change various kinds of HDLs to each other without using hands nor preparing programs at every kind of HDLs.

**SOLUTION:** A syntactic element fetching means 21 calls the HDL template matching the kind of an HDL read from an HDL input device 1 from an HDL template storing section 31, fetches each syntactic element by using the template, and stores the elements in a syntactic element storing section 32. An HDL composing means 22 calls the HDL template matching the kind of an HDL to be changed from the storing section 31 and recomposes an HDL by combining the reserved word of the HDL to be changed with the syntactic element by referring to the HDL template and an HDL output device 4 outputs the recomposed HDL to a display device or another storing medium.

Data supplied from the [esp@cenet](mailto:esp@cenet) database - I2

(d)

(19) 日本国特許庁 (JP)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開 2000-148804

(P 2000-148804A)

(43) 公開日 平成12年5月30日 (2000. 5. 30)

(51) Int. Cl. 7

識別記号

F I

テーマコード (参考)

G 0 6 F 17/50

G 0 6 F 15/60

6 5 4 A 5B046

審査請求 未請求 請求項の数 5

O L

(全 7 頁)

(21) 出願番号

特願平10-315723

(22) 出願日

平成10年11月6日 (1998. 11. 6)

(71) 出願人 000232047

日本電気エンジニアリング株式会社

東京都港区芝浦三丁目18番21号

(72) 発明者 飯塚 卓也

東京都港区芝浦三丁目18番21号 日本電気

エンジニアリング株式会社内

(74) 代理人 100082935

弁理士 京本 直樹 (外2名)

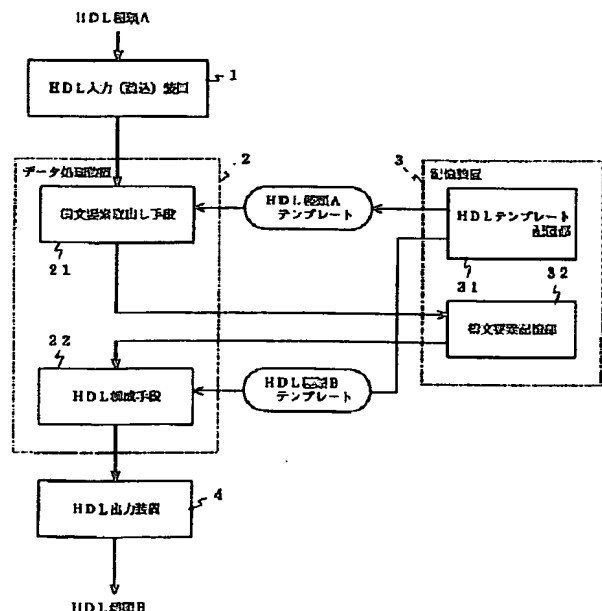
F ターム (参考) 5B046 AA08 BA03 GA01 KA06

(54) 【発明の名称】 HDL仕様変換システム及びそのHDL仕様変換方法並びにその制御プログラムを記録した記録媒体

(57) 【要約】

【課題】 人手を使わず、またHDL種類個別にプログラムを用意することなく、容易に多種のHDL同士を変換可能なHDL仕様変換システムを提供する。

【解決手段】 構文要素抽出手段21はHDL入力装置1から読み込まれたHDLの種類に合ったHDLテンプレートをHDLテンプレート記憶部31から呼出し、そのHDLテンプレートを使用してHDLから各構文要素を抽出し、構成要素記憶部32に格納される。HDL構成手段22は変換したいHDLの種類に合ったHDLテンプレートをHDLテンプレート記憶部31から呼出し、そのHDLテンプレートを参照して変換したいHDLの予約語と構文要素とを合わせてHDLに再構成し、HDL出力装置4によってディスプレイ装置や他の記憶媒体に出力される。



## 【特許請求の範囲】

【請求項 1】 複数種類のハードウェア記述言語各々の構文規則及び予約語と構文要素との位置関係が予め定義された HDL テンプレートを記憶するテンプレートライブラリと、入力されたハードウェア記述言語から前記テンプレートライブラリを参照して前記構文要素を取出す構文要素取出し手段と、前記構文要素取出し手段で取出された構文要素を格納する格納手段と、前記格納手段に格納された前記構文要素を基に変換したいハードウェア記述言語に対応する HDL テンプレートを参照して前記

10 入力されたハードウェア記述言語を前記変換したいハードウェア記述言語に変換する変換手段とを有することを特徴とする HDL 仕様変換システム。

【請求項 2】 前記変換手段は、前記格納手段に格納された前記構文要素と前記 HDL テンプレートの予約語とを合わせて前記変換したいハードウェア記述言語に再構成するよう構成したことを請求項 1 記載の HDL 仕様変換システム。

【請求項 3】 複数種類のハードウェア記述言語各々の構文規則及び予約語と構文要素との位置関係が予め定義された HDL テンプレートを記憶するテンプレートライブラリを参照して入力されたハードウェア記述言語から前記構文要素を取出すステップと、その取出された構文要素を格納するステップと、格納された前記構文要素を基に変換したいハードウェア記述言語に対応する前記 HDL テンプレートを参照して前記入力されたハードウェア記述言語を前記変換したいハードウェア記述言語に変換するステップとを有することを特徴とする HDL 仕様変換方法。

【請求項 4】 前記入力されたハードウェア記述言語を前記変換したいハードウェア記述言語に変換するステップは、格納された前記構文要素と前記 HDL テンプレートの予約語とを合わせて前記変換したいハードウェア記述言語に再構成するようにしたことを請求項 3 記載の HDL 仕様変換方法。

【請求項 5】 コンピュータに、入力されたハードウェア記述言語を変換したいハードウェア記述言語に変換させるための HDL 仕様変換制御プログラムを記録した記録媒体であって、前記 HDL 仕様変換制御プログラムは前記コンピュータに、複数種類のハードウェア記述言語各々の構文規則及び予約語と構文要素との位置関係が予め定義された HDL テンプレートを記憶するテンプレートライブラリを参照して前記入力されたハードウェア記述言語から前記構文要素を取出させ、その取出させた構文要素を格納させ、格納させた前記構文要素を基に前記変換したいハードウェア記述言語に対応する前記 HDL テンプレートを参照して前記入力されたハードウェア記述言語を前記変換したいハードウェア記述言語に変換させることを特徴とする HDL 仕様変換制御プログラムを記録した記録媒体。

## 【発明の詳細な説明】

## 【0001】

【発明の属する技術分野】本発明は HDL 仕様変換システム及びその HDL 仕様変換方法並びにその制御プログラムを記録した記録媒体に関し、特に LSI（大規模集積回路）における論理設計の支援をする支援装置に関する。

## 【0002】

【従来の技術】従来、LSI 設計に使用する HDL [Hardware Description Languages: ハードウェア記述言語（様々な電子回路の構造及び動作を表現するための言語）] には様々な種類が存在する。しかしながら、HDL を入力とする LSI 設計支援プログラムが、全ての種類の HDL に対応している訳ではない。

【0003】HDL に合わせたプログラムを使用する必要があるが、必ずしもそのようなプログラムが存在するとは限らない。そこで、HDL の種類の変換が必要になる。しかも、HDL の内容の可読性、テクノロジーの汎用性等も考慮すると、RT レベル記述 [Register Transfer Level: レジスタ、カウンタ、マルチプレクサ、ALU (Arithmetic and Logic Unit) 等のレベルで機能を表現する記述] での HDL の種類変換が必要となる。

## 【0004】

【発明が解決しようとする課題】上述した従来の異なる種類の HDL 間の変換方法では、人手によって新たに作成し直すか、あるいは特定される 1 種類対 1 種類の変換を直接行うプログラムを実行するしかない。

【0005】また、間接的な方法で実際に論理合成や HDL シミュレーション時に種類に合わせたコンパイラを使用する方法があるが、この方法の場合、HDL 種類に合うコンパイラがあるとは限らず、またコンパイラを通すと RT レベル記述で出力できないゲートレベル [RT レベルよりさらに抽象性の低いレベルであり、アンド (AND)、オア (OR)、インバータ (INVERT) の他、幾つかのフリップ・フロップで機能を表現したもの] になってしまう等の制約がある。

【0006】そこで、本発明の目的は上記の問題点を解消し、人手を使わず、また HDL 種類個別にプログラムを用意することなく、容易に多種の HDL 同士を変換することができる HDL 仕様変換システム及びその HDL 仕様変換方法並びにその制御プログラムを記録した記録媒体を提供することにある。

## 【0007】

【課題を解決するための手段】本発明による HDL 仕様変換システムは、複数種類のハードウェア記述言語各々の構文規則及び予約語と構文要素との位置関係が予め定義された HDL テンプレートを記憶するテンプレートライブラリと、入力されたハードウェア記述言語から前記

テンプレートライブラリを参照して前記構文要素を取出す構文要素取出し手段と、前記構文要素取出し手段で取出された構文要素を格納する格納手段と、前記格納手段に格納された前記構文要素を基に変換したいハードウェア記述言語に対応するHDLテンプレートを参照して前記入力されたハードウェア記述言語を前記変換したいハードウェア記述言語に変換する変換手段とを備えている。

【0008】本発明によるHDL仕様変換方法は、複数種類のハードウェア記述言語各々の構文規則及び予約語と構文要素との位置関係が予め定義されたHDLテンプレートを記憶するテンプレートライブラリを参照して入力されたハードウェア記述言語から前記構文要素を取出すステップと、その取出された構文要素を格納するステップと、格納された前記構文要素を基に変換したいハードウェア記述言語に対応する前記HDLテンプレートを参照して前記入力されたハードウェア記述言語を前記変換したいハードウェア記述言語に変換するステップとを備えている。

【0009】本発明によるHDL仕様変換制御プログラムを記録した記録媒体は、コンピュータに、入力されたハードウェア記述言語を変換したいハードウェア記述言語に変換させるためのHDL仕様変換制御プログラムを記録した記録媒体であって、前記HDL仕様変換制御プログラムは前記コンピュータに、複数種類のハードウェア記述言語各々の構文規則及び予約語と構文要素との位置関係が予め定義されたHDLテンプレートを記憶するテンプレートライブラリを参照して前記入力されたハードウェア記述言語から前記構文要素を取出させ、その取出させた構文要素を格納させ、格納させた前記構文要素を基に前記変換したいハードウェア記述言語に対応する前記HDLテンプレートを参照して前記入力されたハードウェア記述言語を前記変換したいハードウェア記述言語に変換させている。

【0010】すなわち、本発明のHDL仕様変換システムは、様々な種類がありかつ夫々構文規則が異なるHDLにおいて、異種類のHDLであっても同じ意味を持つ構文を構成する要素が同じである点に着目し、HDLの構文要素に着目したデータフィールドをもつ設計回路情報を格納するデータベースと、ある特定種類のHDLの構文規則及びその構文の要素とデータベースとの間の対応規則を書いたテンプレートライブラリと、そのライブラリを参照してHDLからデータベースへの変換を行う第1の変換手段と、同じくライブラリを参照してデータベースからHDLへの変換を行う第2の変換手段とを有している。

【0011】ある種類のHDLはその種類にあったテンプレートライブラリを参照して第1の変換手段が構文毎にその要素を取出し、その要素の情報をデータベースへ格納する。

【0012】このデータベースの内容を基に第2の変換手段が別の種類用のライブラリを参照して構文規則に従って要素を再構成し、その再構成結果を出力する。これによって、結果的にある種類のHDLが別種類のHDLに変換されるので、人手を使わず、またHDL種類個別にプログラムを用意することなく、容易に多種のHDL同士を変換することが可能となる。

#### 【0013】

【発明の実施の形態】次に、本発明の一実施例について図面を参照して説明する。図1は本発明の一実施例によるHDL仕様変換システムの構成を示すブロック図である。図において、本発明の一実施例によるHDL仕様変換システムはHDL入力（読込）装置1と、データ処理装置2と、記憶装置3と、HDL出力装置4とから構成されている。

【0014】HDL入力装置1はHDLを入力するものであり、キーボードや他の記憶媒体からHDLを入力する入力機器からなる。データ処理装置2は構文要素取出し手段21とHDL構成手段22とを持ち、図示せぬ制御メモリのプログラムを基に動作する。尚、制御メモリとしてはROM（リードオンリメモリ）やフロッピーディスク等が使用可能である。

【0015】記憶装置3は予め設定された複数種類のHDLテンプレートを記憶するHDLテンプレート記憶部31と、構文要素取出し手段21で取出された構成要素を記憶する構成要素記憶部32（データベース）とからなる。HDL出力装置4はHDL構成手段22で構成されたHDLをディスプレイ装置（図示せず）や他の記憶媒体に出力する出力機器である。

【0016】ここで、HDLテンプレート記憶部31に予め記憶されたHDLテンプレートには各種HDLの構文規則及び予約語と構文要素との位置関係が予め定義されている。

【0017】HDL入力装置1に読込まれたHDL（例えば、種類A）はデータ処理装置2の構文要素取出し手段21に渡される。構文要素取出し手段21はHDL入力装置1から読込まれたHDLの種類に合ったHDLテンプレート（種類A）をHDLテンプレート記憶部31から呼出し、そのHDLテンプレートを使用してHDL（種類A）から各構文要素を取出す。構文要素取出し手段21によって取出された各構文要素は構成要素記憶部32に格納される。

【0018】構成要素記憶部32に格納された各構文要素はHDL構成手段22に渡される。HDL構成手段22は変換したいHDLの種類（例えば、種類B）に合ったHDLテンプレート（種類B）をHDLテンプレート記憶部31から呼出し、そのHDLテンプレートを参照してHDL（種類B）の予約語と構文要素とを合わせてHDL（種類B）に再構成する。HDL構成手段22で再構成されたHDL（種類B）はHDL出力装置4によ

ってディスプレイ装置や他の記憶媒体に出力される。

【0019】図2は本発明の一実施例によるHDL仕様変換システムの動作を示すフローチャートであり、図3は図2のデータベースのフォーマットの一例を示す図であり、図4は本発明の一実施例によるHDL→データベース変換の一例を示す図であり、図5及び図6は本発明の一実施例によるデータベース→HDL変換の一例を示す図である。

【0020】図3において、このデータベースのレコード(行)はHDLの構文用途(意味)で分けられ、フィールドは” ; ” で区切られ、その構文の要素が格納されている。

【0021】ここで、numは識別番号を、wordは構文を表すキーワード(例えば、レジスタの場合、register)を、\$1はクロック信号名を、\$2はクロックエッジ(立上りまたは立下り)を、\$3は出力信号名ービット範囲を、\$4は入力信号名ービット範囲を夫々示している。但し、\$1以降はキーワードによって意味が変わる。

【0022】テンプレートライブラリでは構文種類と、その構文種類がそのHDL種類でどのような構文規則で記述されているかが定義されている。構文規則とはそのHDL種類での予約語と構文要素とそれらの位置関係とを示す。テンプレートライブラリはHDLを読込む場合、あるいはHDLを出力する場合に同じものを使用する。例えば、HDL(種類A)を読込む場合と、HDL(種類A)で出力する場合とは同じHDL(種類A)用のテンプレートライブラリを使用する。

【0023】構文要素を取出す場合には予約語との位置関係をテンプレートで見て、定義されている位置にあるものを構文要素として抽出する。HDLに出力する場合には構文要素を定義されている位置に入れ、予約語と合わせて全て出力する。HDL(種類A)の例[図4の(1)参照]でいえば、register//の後に書かれているのがHDL(種類A)でのレジスタ構文の規則になる。

【0024】” REG”、” IF”、” THEN” はHDL(種類A)の予約語であり、” REG” の次の位置にあるものが「\$3=レジスタ構文の出力信号名」、” IF” の次の位置にあるものが「\$1=レジスタ構文のクロック信号名」という構文要素になる。

【0025】テンプレートライブラリはHDL間で同意の構文要素をデータベース中の同じレコードと対応するようにして作成される。HDL(種類A)で書かれた論理はHDL(種類A)用のテンプレートライブラリ31aを参照して構文要素取出し手段21によって読込まれ、構文要素が取出される(図2ステップS1)。取出された構文要素はデータベース5(構成要素記憶部32)に出力される(図2ステップS2)[出力例は図4の(2)参照]。

【0026】図4の(2)はHDL(種類A)でのレジスタを表す構文であり、HDL(種類A)のテンプレートライブラリ[図3の(1)参照]のレジスタ構文部中の「\$1」の箇所にあたる要素” CLK” がクロック信号名として取出される。同様に、「クロックエッジ(\$2)」として「立上がり」、「出力信号名(\$3)」として” PS(0)~(3)”、「入力信号名(\$4)」として” D(0)~(3)” がテンプレートライブラリを参照して要素として取出され、データベースに出力される[図4の(3)参照]。

【0027】データベースに出力された各要素はデータベースの読込みによって同意のレコードから取出され、別種類のテンプレートライブラリ31bを参照してHDL構成手段22によってライブラリ中の記述仕様に基いて出力される(図2ステップS3)。

【0028】データベース[図5の(1)参照]でレジスタ構文の\$1のフィールド” CLK” はHDL(種類B)のテンプレートライブラリ[図5の(2)参照]を参照した場合、” レジスタ構文” \$1の箇所に出力される。\$2、\$3、\$4も夫々HDL(種類B)のテンプレートライブラリで定義されている箇所に出力される。その結果、入力されたHDL(種類A)はHDL(種類B)に変換されてHDL出力装置4から出力される(図2ステップS4)[図5の(3)参照]。

【0029】同様に、HDL(種類C)のテンプレートライブラリを使用すれば、HDL(種類C)での” レジスタ構文” で出力される(図6参照)。

【0030】このように、構文要素取出し手段21が入力されたHDL(種類A)に対応するHDL(種類A)テンプレートライブラリ31aを参照して構文毎にその要素を取出し、その要素の情報を構成要素記憶部32(データベース5)に格納し、HDL構成手段22が構成要素記憶部32(データベース5)に格納された構成要素を基に、HDL(種類B)テンプレートライブラリ31bを参照して構文規則に従って要素を再構成し、その再構成結果をHDL出力装置4から出力することによって、人手を使わず、またHDL種類個別にプログラムを用意することなく、容易に多種のHDL同士を変換することができる。また、HDLだけでなく、ゲートレベルのネットリストでも同様に多種類間の変換を容易に行うことができる。

【0031】

【発明の効果】以上説明したように本発明によれば、複数種類のハードウェア記述言語の構文規則及び予約語と構文要素との位置関係が予め定義されたHDLテンプレートを記憶するテンプレートライブラリを参照して入力されたハードウェア記述言語から構文要素を取出し、その取出された構文要素を格納し、格納された構文要素を基に変換したいハードウェア記述言語に対応するHDLテンプレートを参照して入力されたハードウェア記述言

語を変換したいハードウェア記述言語に変換することによって、人手を使わず、またHDL種類個別にプログラムを用意することなく、容易に多種のHDL同士を変換することができるという効果がある。

【図面の簡単な説明】

【図1】本発明の一実施例によるHDL仕様変換システムの構成を示すブロック図である。

【図2】本発明の一実施例によるHDL仕様変換システムの動作を示すフローチャートである。

【図3】図2のデータベースのフォーマットの一例を示す図である。

【図4】本発明の一実施例によるHDL→データベース変換の一例を示す図である。

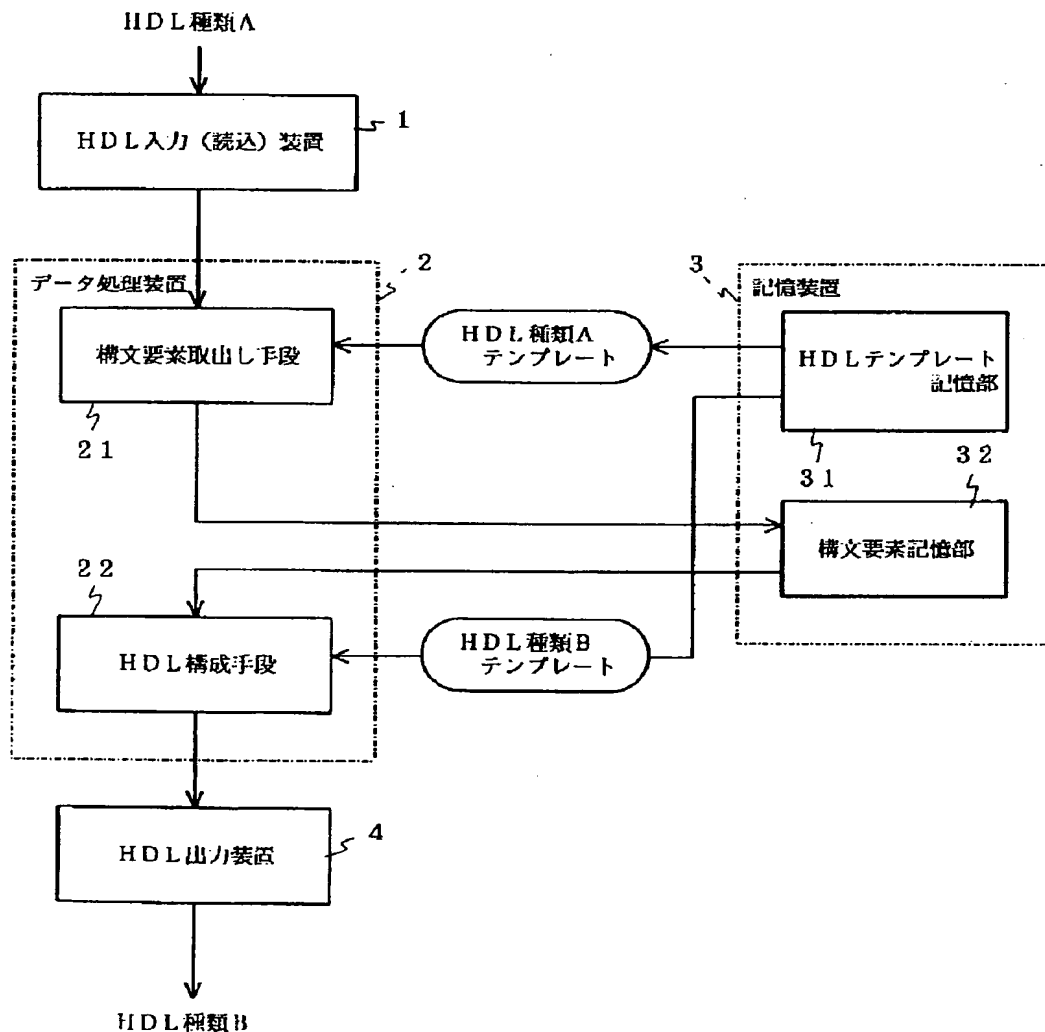
【図5】本発明の一実施例によるデータベース→HDL変換の一例を示す図である。

【図6】本発明の一実施例によるデータベース→HDL変換の一例を示す図である。

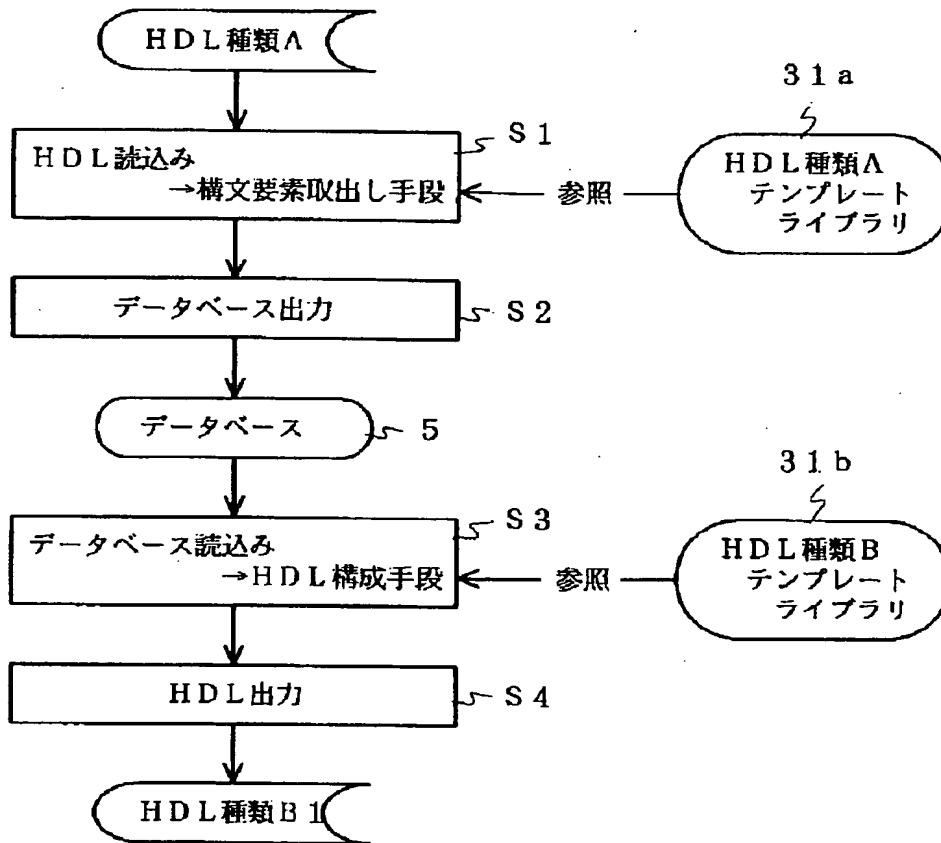
【符号の説明】

- 1 HDL入力（読込）装置
- 2 データ処理装置
- 3 記憶装置
- 4 HDL出力装置
- 5 データベース
- 21 構文要素取出し手段
- 22 HDL構成手段
- 31 HDLテンプレート記憶部
- 31a HDL種類Aテンプレートライブラリ
- 31b HDL種類Bテンプレートライブラリ
- 32 構成要素記憶部

【図1】



【図2】



【図3】

num;word;\$1;\$2;\$3;\$4  
 num : 識別番号  
 word : 構文を表すキーワード (レジスタの場合 register)  
 \$1 : クロック信号名  
 \$2 : クロックエッジ (立ち上がり or 立ち下がり)  
 \$3 : 出力信号名 ビット範囲  
 \$4 : 入力信号名 ビット範囲;     \$1からは、キーワードにより意味が変わる。

【図4】

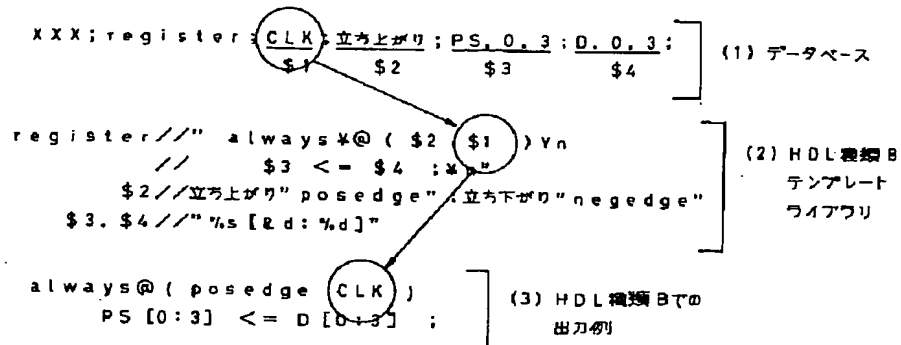
```

register // " REG $3 = IF $1 $2 THEN $4 ; #n" ] (1) HDL種類A
          $2 // 立ち上がり, UP, " : 立ち下がり, DN. "   テンプレート
          $3, $4 // "%s (%d-%d)"                      ライブラリ

REG PS (0-3) = IF CLK . UP. THEN D (0-3) ; ] (2) HDL種類A
                                                記述例

XXX; register : CLK; 立ち上がり; PS, 0, 3; D, 0, 3; ] (3) データベース
               $1      $2      $3      $4      出力例
  
```

【図5】



【図6】

